

1 Claims 1-2, 8-11, 13, 17-20, 22-23 and 26-28 are amended:

2
3 **1. (Currently Amended)** A method for testing at least one
4 software application, the method comprising:

5 installing at least one hook function into an application programming
6 interface (API) of an operating system, the at least one hook function
7 configured to monitor operating system messages communicated with the
8 software application during execution of the software application being tested;

9 retrieving information descriptive of a state of operation of the software
10 application being tested and at least one graphics element rendered during
11 ~~deterministic~~ execution of the software application being tested, wherein the
12 information identifies an executable feature associated with the at least one
13 graphics element, and wherein at least some of the retrieved information
14 descriptive of the state of operation is based on messages monitored by way of
15 the at least one hook function;

16 storing information related to an association between the executable
17 feature and the at least one graphics element and the state of operation of the
18 software application in a map data structure containing information related to at
19 least one graphics element for testing, the association and information being
20 stored in the map data structure during execution of the software application
21 being tested;

22 automatically selecting an executable feature from the map data
23 structure based on the association stored in the map data structure, ~~wherein~~
24 ~~selecting the executable feature proceeds according to a sequence determined~~
25

1 by one of a plurality of deterministic modes for a systematic order of software
2 application execution during testing;

3 automatically executing the selected executable feature associated with
4 the graphics element; and

5 dynamically updating the information related to the state of operation of
6 the software application and the association in the map data structure upon
7 execution of the executable feature.

1 **2. (Currently Amended)** The method of claim 1, further
2 comprising:

3 dynamically updating information in the map data structure descriptive of
4 at least one second graphics element resulting from the exposure of a new state
5 of operation of the software application in response to the execution of the
6 executable feature;

7 displaying the at least one second graphics element;

8 retrieving information descriptive of the state of operation of the software
9 application being tested and the at least one second graphics element rendered
10 during execution of the software being tested, the information including a second
11 executable feature associated with the at least one second graphics element and the
12 new state of operation of the software application;

13 storing the information descriptive of the new state of operation of the
14 software application being tested and the second executable feature in association
15 with the at least one second graphics element in the map data structure during
16 execution of the software being tested; and

17 automatically executing the at least one second executable feature stored
18 in association with the second graphics element.

19
20 **3. (Previously Presented)** The method of claim 1 wherein the
21 retrieving comprises capturing information pertaining to the graphics element and
22 the state of operation of the software application being tested.

1 **4. (Previously Presented)** The method of claim 1, wherein the
2 storing includes updating an indicator associated with the at least one graphics
3 element when the executable feature stored in association with the at least one
4 graphics element is executed.

5
6 **5. (Previously Presented)** The method of claim 1 wherein the
7 storing includes organizing the retrieved information such that the executable
8 feature stored in association with the at least one graphics element can be
9 interpreted by a computer-executable application capable of accessing the
10 retrieved information.

11
12 **6. (Previously Presented)** The method of claim 1 wherein the
13 storing includes organizing the retrieved information such that the executable
14 feature stored in association with the at least one graphics element and the state of
15 operation of the software application being tested can be interpreted by a user
16 capable of accessing the retrieved information from memory.

17
18 **7. (Canceled).**

19
20 **8. (Currently Amended)** The method of claim 1 wherein the
21 automatically selecting comprises automatically selecting an executable feature
22 not previously executed.
23
24
25

1 **9. (Currently Amended)** The method of claim 8 wherein the
2 automatically selecting comprises reviewing an indicator to automatically select an
3 executable feature not previously executed.

4
5 **10. (Currently Amended)** The method of claim 1 wherein the
6 automatically selecting comprises automatically selecting executable features in a
7 depth-first mode of operation.

8
9 **11. (Currently Amended)** The method of claim 1 wherein the
10 automatically selecting comprises automatically selecting executable features in a
11 breadth-first mode of operation.

12
13 **12. (Canceled).**
14
15
16
17
18
19
20
21
22
23
24
25

1 **13. (Currently Amended)** A system for generating a map,
2 comprising:

3 a capture agent for retrieving information descriptive of a state of
4 operation of a software application being tested and a plurality of graphics
5 elements rendered during ~~deterministic~~ execution of the software application, the
6 information including an executable feature associated with each graphics
7 element, the capture agent configured to install at least one hook function into an
8 application programming interface (API) of an operating system, wherein the at
9 least one hook function is configured to monitor messages communicated
10 between the operating system and the software application during execution of
11 the software application being tested;

12 an application driver for storing information in a map data structure
13 related to an association between each executable feature and corresponding
14 graphics element and a state of operation of the software application during
15 execution of the software application being tested, wherein the map data
16 structure contains information related to at least one graphics element for testing;

17 an indicator for tracking a dynamic updating of the information an
18 application driver for ~~deterministically~~ automatically selecting one of the
19 executable features stored in the map data structure based on the information
20 stored in the map data structure, ~~wherein deterministically selecting proceeds~~
21 ~~according to a sequence determined by one of a plurality of deterministic modes~~
22 ~~for a systematic order of software application execution during testing;~~

23 a command agent for automatically executing the selected executable
24 feature; and
25

an indicator for tracking a dynamic updating of the information related to the association and the state of operation of the software application in the map data structure upon the automatic execution of the selected executable feature.

14. (Original) The system of claim 13 wherein the capture agent is invoked by the application driver.

15. (Original) The system of claim 13 wherein the capture agent submits the retrieved information to the application driver.

16. (Canceled).

17. (Currently Amended) The system of claim 13, wherein the application driver ~~deterministically~~ automatically selects one of the executable features that has not been previously executed.

18. (Currently Amended) The system of claim 13, wherein the application driver reviews the indicator to automatically select the one executable feature.

19. (Currently Amended) The system of claim 13, wherein the application driver ~~deterministically~~ automatically selects executable features according to a depth-first deterministic mode of operation.

1 **20. (Currently Amended)** The system of claim 13 wherein the
2 application driver ~~deterministically~~ automatically selects executable features
3 according to a breadth-first deterministic mode of operation.
4

5 **21. (Canceled).**
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

22. (Currently Amended) A method for systematically invoking an executable feature of a software application having a graphical user interface, the method comprising:

installing at least one hook function into an application programming interface (API) of an operating system, the at least one hook function configured to monitor operating system messages communicated with the software application during execution of the software application being tested;

retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered during deterministic execution of the software application, the information including an executable feature associated with the at least one graphics element, at least some of the retrieved information descriptive of a state of operation of the software application retrieved by way of messages monitored by the at least one hook function;

storing information related to an association between the executable feature and corresponding graphics element and the state of operation of the software application in a map data structure to contain information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application;

automatically selecting from the map data structure at least one executable feature associated with a graphics element that has not been previously executed, wherein selecting the at least one executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing; and

automatically executing the selected at least one executable feature.

1 **23. (Currently Amended)** The method of claim 22 further
2 comprising, in response to executing the selected executable feature:

3 dynamically updating the information related to the association and the
4 state of operation of the software application in the map data structure upon
5 execution of the selected at least one executable feature;

6 displaying a second graphics element;

7 retrieving information descriptive of the second graphics element
8 rendered during execution of the software application and the state of operation of
9 the software application being tested, the information including an association of
10 a second executable feature with the second graphics element and a new state of
11 operation of the software application;

12 storing the information descriptive of the new state of operation of the
13 software application and the association between the second executable feature
14 and the second graphics element in the map data structure;

15 automatically selecting from the map data structure an executable feature
16 that has not been previously executed, according to a sequence determined by
17 one of a plurality of deterministic modes of execution of the software
18 application; and

19 automatically executing the selected executable feature.
20

21 **24. (Previously Presented)** The method of claim 22, wherein the
22 retrieving comprises capturing information pertaining to the second graphics
23 element.
24
25

1 **25. (Previously Presented)** The method of claim 22, wherein the
2 storing comprises updating an indicator associated with the second graphics
3 element when an executable feature stored in association with the graphics
4 element is executed.

5
6 **26. (Currently Amended)** The method of claim 22, wherein the
7 automatically selecting comprises reviewing an indicator to determine an
8 executable feature not previously executed.

9
10 **27. (Currently Amended)** The method of claim 22, wherein the
11 automatically selecting ~~one of the plurality of deterministic modes of execution of the~~
12 ~~software application~~ includes a depth-first mode of operation.

13
14 **28. (Currently Amended)** The method of claim 22, wherein the
15 automatically selecting ~~one of the plurality of deterministic modes of execution of~~
16 ~~the software application~~ includes a breadth-first mode of operation.

17
18 **29-39. (Canceled).**
19

20 **40. (Previously Presented)** The method of claim 1, wherein a state
21 of operation of the software application includes a distinctive set of graphic
22 elements, content, and associated actions of the software application during
23 execution.
24
25